ARC

独立看门狗

支持	www.armrunc.com http://armrunc.taobao.com

Change History

版本号	更改描述	作者	修改时间
1.0.0	第一版	armrunc	2012-3-6

目录

第7章 独立看门狗

7.1 独立看门狗简介

看门狗的功能是：它可以让 STM32 在意外状况下（比如软件陷入死循环，跑飞，硬件遇到强干扰）重新回复到系统上电状态，以保证系统出问题的时候重启一次，就像手动按复位键一样。

STM32 看门狗设计上就是一个计数器，计数器能装载的最大数值和计数器的位数有关，一旦开启看门狗，它就开始不停的数机器周期，数一个机器周期计数器就减 1，直到计数器计数为 0 就产生一个复位信号，重启系统。

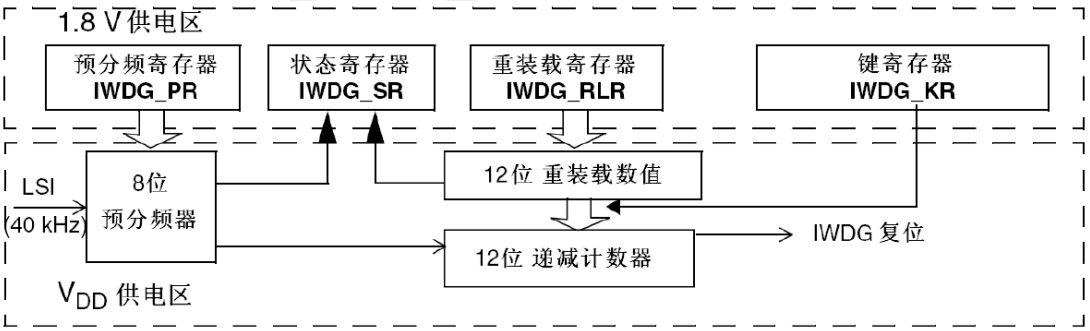
我们在设计程序时，根据你目标的重启时间和看门狗计数器的时钟周期算一下计数器的初始值，在这个初始值递减到 0 之前，你重新装载计数器（这个过程叫“喂狗”），STM32 不会重启，而由于软件或硬件的原因，导致在计数器递减到 0 之前无法喂狗，STM32 将会重启。

独立看门狗(IWDG)由专用的低速时钟(LSI)驱动，即使主时钟发生故障它也仍然有效。它最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的场合。

独立看门狗(IWDG)主要性能：

- 自由运行的递减计数器
- 时钟由独立的 RC 振荡器提供(可在停止和待机模式下工作)
- 看门狗被激活后，则在计数器计数至 0x000 时产生复位

它的的方框图如下：



7.2 独立看门狗应用实例 ----- 按键重载看门狗计数器

7.2.1 实例描述

本实例先初始化 LED 按键和 SysTick，然后初始化独立看门狗，设置的超时时间大约为 2s，在此时间内如用户按下按键 0 或者按键 1，看门狗会重新装载计数器，防止系统重启，若用户没有按任意一个键，系统将会重启。为了使开发者有个直观印象，使用 LED0 和 LED1 指示，一开机首先保持 LED0 和 LED1 熄

灭 1s,然后点亮这两个 LED，所以，如果你在 2s 之内一直有按下按键动作，此两个 LED 灯将一直保持亮着状态，反之，系统重启，会看到有 1s 的熄灭状态。

7.2.2 硬件设计

参见 LED 和 EXTI 这两章节。

7.2.3 软件设计

本实例首先初始化 LED0，LED1，用来指示系统的运行状态，然后在初始化按键 0 和按键 1，用来喂狗，然后在初始化 SysTick 用来精确延迟，再初始化独立开门狗，在外部中断 2 和外部中断 3 内喂狗。

文件 IWDG_main.c：

```
/**
   * @brief   Main program, IWDG example.
   * @param   None
   * @retval None
   */
int main(void)
{
      ARC_LED_Init();
      ARC_Button_Init();
      ARC_SysTick_Init();
      ARC_IWDG_Init(IWDG_Prescaler_256, (2 * 40000 / 256));
      ARC_LED_Set(0, 1);
      ARC_LED_Set(1, 1);
      ARC_SysTick_Delay(1000);
      ARC_LED_Set(0, 0);
      ARC_LED_Set(1, 0);
      IWDG_Enable();
      while (1)
      {
      }
}
```

文件 ARC_IWDG.c：

```
/** @defgroup ARC_IWDG_Private_Functions
   * @{
   */

/**
```

```
  * @brief   Initialize IWDG.
  * @param   None
  * @retval None
  */
void ARC_IWDG_Init(uint8_t IWDG_Prescaler, uint16_t Reload)
{
    /* IWDG timeout equal to (Reload / (40KHz(LSI) / IWDG_Prescaler)) ms
      (the timeout may varies due to LSI frequency dispersion) */
    /* Enable write access to IWDG_PR and IWDG_RLR registers */
    IWDG_WriteAccessCmd(IWDG_WriteAccess_Enable);

    /* IWDG counter clock: 40KHz(LSI) / IWDG_Prescaler */
    IWDG_SetPrescaler(IWDG_Prescaler);

    /* Set counter reload value to Reload */
    IWDG_SetReload(Reload);

    /* Reload IWDG counter */
    IWDG_ReloadCounter();
}
```

文件 stm32f10x_it.c：

```
/**
  * @brief   This function handles External interrupt Line 2 request.
  * @param   None
  * @retval None
  */
void EXTI2_IRQHandler(void)
{
    if(EXTI_GetITStatus(EXTI_Line2) != RESET)
    {
        IWDG_ReloadCounter();
        EXTI_ClearITPendingBit(EXTI_Line2);
    }
}

/**
  * @brief   This function handles External interrupt Line 3 request.
  * @param   None
  * @retval None
  */
void EXTI3_IRQHandler(void)
{
    if(EXTI_GetITStatus(EXTI_Line3) != RESET)
```

```
    {
        IWDG_ReloadCounter();
        EXTI_ClearITPendingBit(EXTI_Line3);
    }
}
```